International Journal of Communication and Information Technology

E-ISSN: 2707-6628
P-ISSN: 2707-661X
www.computersciencejournals.com/ijcit
IJCIT 2023; 4(1): 01-11
Received: 01-10-2022
Accepted: 04-11-2022

Md Ibrahim
Department of Computer Science and Engineering, World University of Bangladesh, Dhaka, Bangladesh

Corresponding Author:
Md Ibrahim
Department of Computer Science and Engineering, World University of Bangladesh, Dhaka, Bangladesh

Design and development of a decentralized voting system using blockchain

Md Ibrahim

DOI: https://doi.org/10.33545/2707661X.2023.v4.i1a.53

Abstract
E-voting reduced the cost of election and provided convenience to some extent as compared to the traditional approach of pen and paper but it was considered to be unreliable as anyone having access to the machine physically can obstruct the machine and alter the votes. Also, in order to control the entire procedure from electronic voting to electoral results and tracking the outcomes, a central system is required. Voters are not completely secure as vote can be targeted easily. It also possesses a great threat to the right to vote and transparency. For long, different e-voting systems have been provided with the goal of increasing security and minimizing cost. With the launch of Ethereum, a decentralized platform which runs decentralized applications on it, a secured voting system now seems possible.
Making a project successful requires hard work and study. For study purposes, some previous works have been studied and gathered a lot of information which makes the project possible. The job portal-based application and some other things have been learned from the study.
The methodology is a very important process for a project. Without this process, a project can't build up properly. It is a blueprint of a project. This project has used a static waterfall methodology which has a feedback path from one phase to its existing phase.
It will allow accounts connected to the network to vote in the election and all votes come in the real time. Application's logical code will deploy in smart contract to a local Ethereum blockchain. It will have a test-driven development (TDD) part, means tests against the smart contract to ensure that the code is bug free. It will also have a client-side application that will allow users connected to the network to interact with the smart contract and vote in the election. User-Interface will develop using HTML, CSS, Java Script.
This paper provides a solution for removing inconveniences from conventional elections using blockchain that has emerged as an exciting technology for various application due to its unique characteristics that outperform other technologies. The goal of this research is to establish a system for e-voting that is decentralized rather than centralized by using blockchain technology that guarantees protection to electorate's identity, data transfer privacy and variability by an open and transparent voting process.

Keywords: E-voting, decentralized, blockchain, phase

Introduction
In a centralized system, A web Application depends on a central server. All the code to that web application lives on this central server, and all the data lives in a centralized database. That means the organizing authority can access and change code, data, and the system's rules at any time.
The blockchain is a peer-to-peer network of nodes. Each of the nodes will get a copy of all the data that is shared across the blockchain. All this data is connected in the bundles of records called blocks, chained together to create the public ledger. All the nodes on the network work together to ensure that all the data on the general legend remain secure and unchanged. On the blockchain, all the data doesn't lie on a central server; instead, the data is decentralized. It's distributed across every node connected to the network. The device connected to the blockchain is a node.
Blockchain technology is decentralized, distributed, immutable, irreversible and provides data security. These technological features operate through advanced cryptography, providing a security level more significant than any previously known database. The ledger is shared through every node connected to the network. That means data and logic are transmitted through the connected node. So, Blockchain can be referred to as a decentralized public database with replicates distributed over several nodes simultaneously.

In Blockchain, there is no authority in charge of managing and maintaining the ledger of transactions. Blockchain technology allows a secure validation of a transaction's data integrity.

This project provides a solution for removing inconveniences from conventional elections using blockchain. That has emerged as an exciting technology for various applications due to its unique characteristics that outperform other technologies. This project aims to establish a decentralized system for e-voting rather than a centralized one by using blockchain technology that guarantees the protection of the electorate's identity, data transfer privacy, and variability through an open and transparent voting process.

Objectives

1. To study the security challenges associated with electronic voting systems and identify requirements.
2. To design and develop an electronic voting system
3. Implement a Ethereum smart contracts and decentralized apps underlying the particular case study in the scope of this project.

Justification of Study

Voting is a democratic practice of a group to share their opinions, decide, and express their choice against debates and controversies like ballot questions, candidate elections, political parties, etc. The traditional voting system is based on pen and paper, which fails in some security concerns. It isn't traceable and verifiable the whole voting process. The results of voting events have always been questionable and not trustworthy to voters. The electronic voting system is a new election system that has become popular. Voting online pull's in different controversies also gained popularity. But the organizing authority on anyone with physical access to such a system or server can hack or alter the votes, thereby affecting all votes cast on this system.

So, the election's fundamental rules get broken, and voters' freedoms and human rights are violated. The problem with the current ballot voting system is being taken advantage of by people or organizations looking to gain power. The current ballot system does offer anonymity to the voter, but the counting process is not transparent. People are supposed to trust the result provided by an Election commission or a government body. This makes the process of counting a significant vulnerability in the current process. There are also major electoral scams such as voter fraud, ballot stuffing and booth capturing. All these make it very difficult for organizers of an election to distinguish between the actual votes and votes added without authorization.

To solve the problem, we use Blockchain technology, which is decentralized, distributed, immutable, irreversible, and provide data security. We will develop a decentralized voting platform based on the Ethereum blockchain. It allows accounts connected to the network to vote in the election, and all votes come in real-time.

Scope of Study

Electoral systems are comprised of a set of rules that determine how elections should be conducted and how their results are decided. These rules encapsulate all aspects of the voting process, such as when an election is to take place, who can vote, who can stand as a candidate, how the ballots are to be labeled and cast, and how the votes are to be counted, among others. To illustrate an example, in a traditional voting system, a person of age would attend their parish or consular district, provide their identification card to a polling agent and receive a ballot paper with a checkbox list of the possible candidates to vote for. The person in question proceeds to an isolated area, where they may select a single candidate or none from the list with a checkmark. The ballot is then delivered to the ballot box and counted by a teller once the deadline for voting finishes. After all the votes are counted, the results are calculated and made available to the public by the government by the many tellers across the country. An electronic voting system that mimics real-world scenarios has long been desired. However, up until this point, it had not been possible to fully address all the mandatory properties of a real-world voting scheme simultaneously. These properties are, among others, security, anonymity, coercion resistance and non-traceable. Recently, with the onset of new technologies and research, however, it is not only possible to fulfill these same properties but also to improve the anonymity and convenience of voting. One embodiment of such technology is the Ethereum Blockchain.

It possesses the following properties

- It is practically immutable, as altering information would require copious amounts of computing power;
- It is transparent, as anyone can view its data;
- It is decentralized, as there is no central authority behind it;
- It is used to build smart contracts and Dapps.

Literature Review

Review of Relevant Literature

In their study discussed a number of challenges in e-voting system including legal challenges, social and cultural challenges, technical challenges, attacks, etc (E. Elewa, A. AlSammak, A. AbdElRahman, T. ElShishtawy, 2015) [4].

In their work identified an experiment to test the validity of election results and to enhance transparency and voter participation within electronic elections. This proposal was based upon two aspects i.e. distributed collection of pole tape, made by mobile devices by voters and crowdsourcing election data verification by electoral authority (Aranha DF, Ribeiro H, Paraense ALO, 2016) [1].

If voters make use of voting protocol correctly then there will be no chance of attack on results of elections and they give a statistical method to improve the security of e-voting (Gjøsteen K, Lund AS, 2016) [5].

Paper explores how to properly develop voting machine interfaces to promote the role of electors and electoral administrators and then use such interfaces for complex elections. The problem of relying on a remote voting device is said to be firmly linked to the interface provided which in turn influences votes as verification of voting is an important issue. Much of the work in remote– electronic voting involves cryptography voting protocols design and verification to safeguard desired property (Budurushi J, Renaud K, Volkamer M, Woide M, 2016) [3].

In order to make specific recommendations on the type of voting system that is best suited to that particular context, they introduced a model for the comparison of voting schemes in any given electoral setting and the model was applied to the specific context of Estonian internet voting. (Neumann S, Volkamer M, Jurlind B, Prandrini M, 2016) [8].

As there were various problems with electronic voting especially related to physical security, people began to look for solutions to the problems that's when blockchain came into the field of e-voting, initially blockchain was used for bitcoin. This work discusses how to take the advantages of blockchain technology in the process of e-voting to make it safe, secure, anonymous, etc (Ayed, A.B., 2017) [2].

In their work make use of smart contracts in decentralized blockchain technology for e-voting to engage all voters in evaluating and recording ballots. It increases the trust of electorate and decreases the misuse of election capital (Hsiao JH, Tso R., Chen CM., Wu ME., 2018) [6].

In their study used NetVote for user interface of the program, it uses decentralized application. The dApp admin helps electoral administrators to decide electoral policy, generate voting, register rules, opening and closing of voting. Identification of voters is done by other applications like biometric readers. To test and check the results of election TallydApp is used. This NetVote reinforce three kinds of elections: private election, open election, Token holder elections (Jonath Alexander, Steven Lander, Ben Howerton, 2018) [7].

Methodology

The waterfall model interprets the software development process in a linear, sequential flow. Software development life cycle (SDLC) models have been created like a waterfall, fountain and fix, spiral build and rapid prototyping, incremental, and synchronize and stabilize. Those are the oldest models, and the waterfall model is the best known. The waterfall approach was the first SDLC Model to be used in Software Engineering to ensure the project's development. The Software Development Life Cycle provides system designers and developers to follow a sequence of activities. A Software Development Life Cycle (SDLC) agglutinates to essential phases for developers, such as planning, analysis, designs, and implementation, and are explained in the section below.

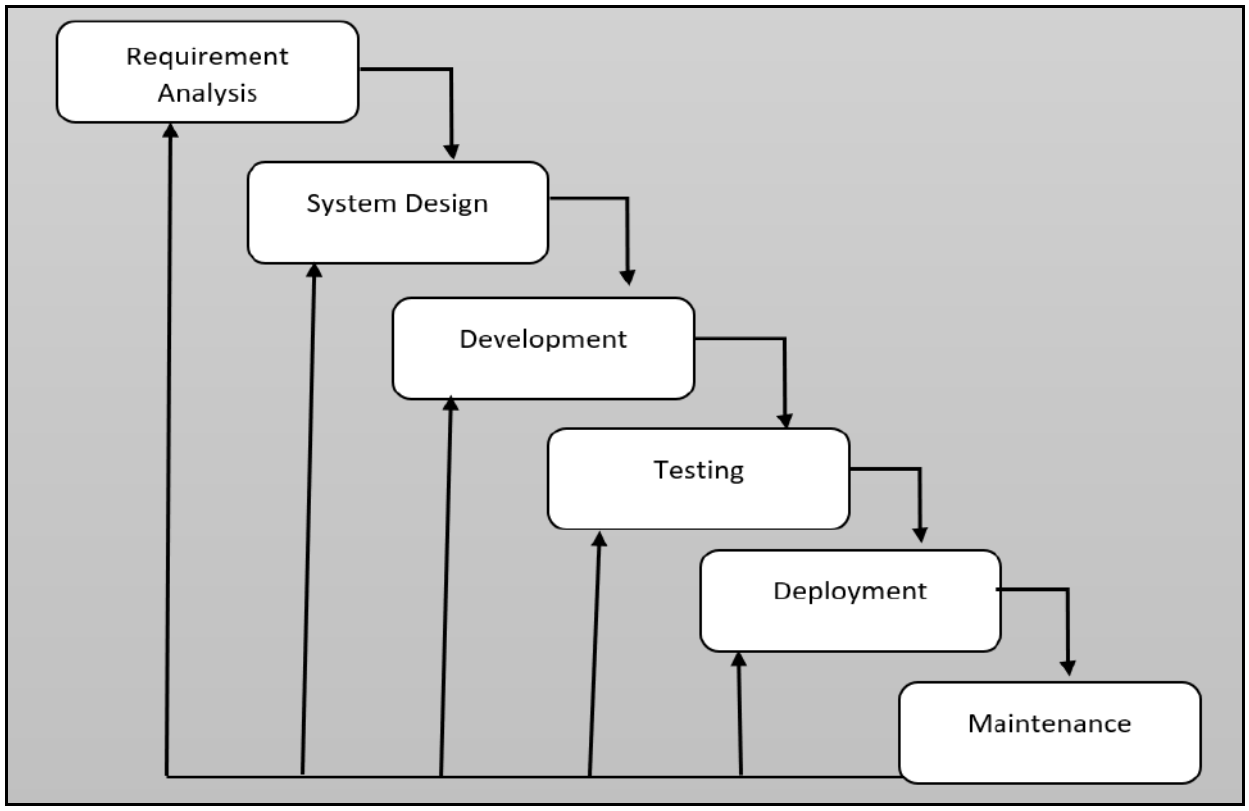

Fig 1: Waterfall Model (Winston Walker Royce, 1970)

Description of Methodology

There are several steps to take an overview of our production house website. They are given below-

Requirement Analysis: This phase captures all possible needs for the system to be developed and documents them in a requirement specification document. We have reviewed many specifications of our project. We identified the hardware and software required for the development by using the waterfall model.

System Design: The goal of this phase is to convert the requirements gathered into a form that can be coded in a programming language. It includes high-level and detailed design as well as the overall software architecture. We created many diagrams to construct an efficient development model, those are use case diagram, data flow diagram and entity relationship diagram.

Development: With inputs from the system design, the system is first code and developed in small programs called units, which are integrated into the next phase. We used Visual Studio Code, Dependency NPM (Node Package Manager), Truffle framework, Ganache, Metamask, React, and Coding language- (Solidity, HTML, CSS, JavaScript).

Testing: In this phase, the software will be tested for its functionality, and if any bug is found, make sure that the bug is fixed. First, we will do the unit test for each unit after development. When the unit test is completed, we will

integrate each department into a system and do the integration test. We can ensure that the web-based system is functioning correctly and can be accepted by real-time users by performing website testing. If any bug is found, we will try to fix it.

Deployment
The functional and non-functional testing is done, and the product is deployed in the customer environment or released into the local and global market.

Maintenance
Some issues come up in the client environment. To fix those issues, patches are released. Also, to enhance the product, some better versions are released. Maintenance is done to deliver these changes in the customer environment.

Justification of Methodology
For our project methodology, we followed the waterfall model because this model is simple and easy to understand and use. It is easy to manage due to the rigidity of the model because each phase of our project has specific deliverables and a review process. In this model, phases are processed and completed one at a time. Phases do not overlap. It allows correcting the errors that are committed, and these changes are reflected within the later phases. Requirements are very well documented, cleared, and fixed. Coding and testing go much faster because they follow to complete small items to code and test. Risk Management and Testing are easy. The waterfall model works well for our project because our requirements are clearly defined and very well understood.

Requirement Analysis & Design
System Requirement
This project has two system requirement, Hardware and Software.

Software Requirements
1. Dependency NPM (Node Package Manager)
2. Truffle framework
3. Ganache
4. Metamask
5. Coding language; solidity, HTML, CSS, JavaScript.
6. Windows (OS).

Hardware Requirement: 64-bit hardware

User Requirements
- Add candidate.
- Import voter account.
- Account authentication.
- Voter can vote once
- Win with maximum vote.

System Design
Flow chart
A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task.

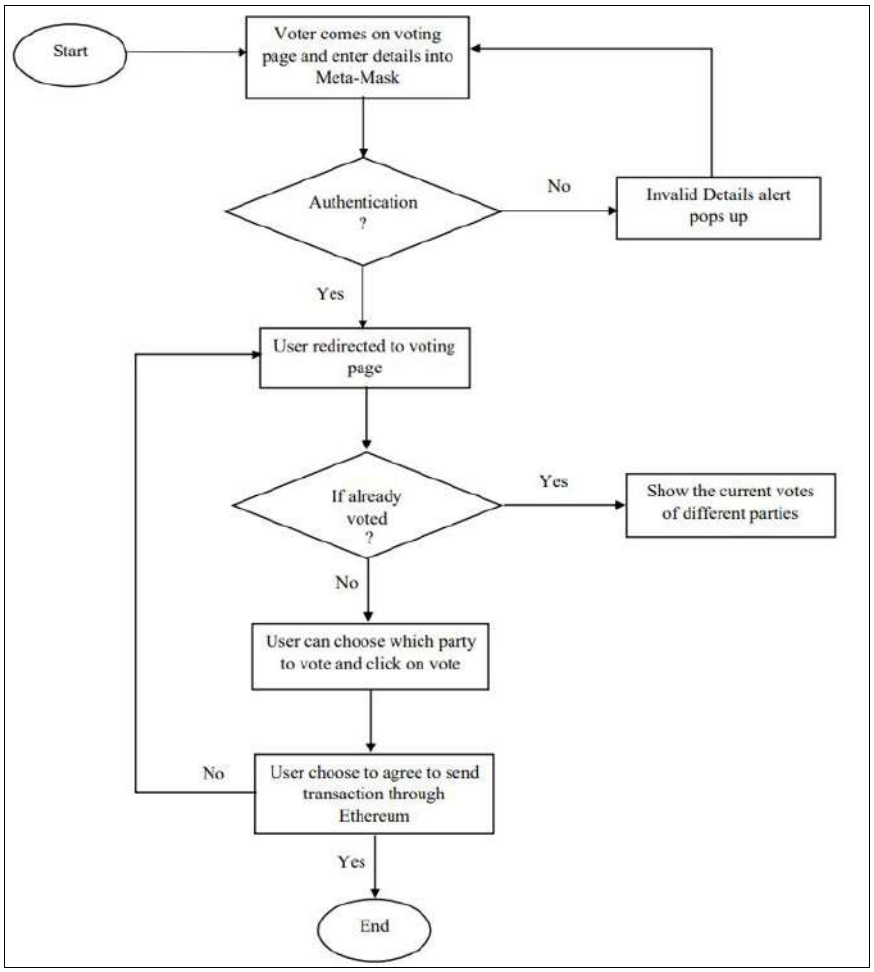

Fig 2: Flow chart

Use Case Diagram: The use case diagram is a way to summarize details of our system and the users within our system. It is generally shown as a graphic depiction of interactions among different elements in our system.

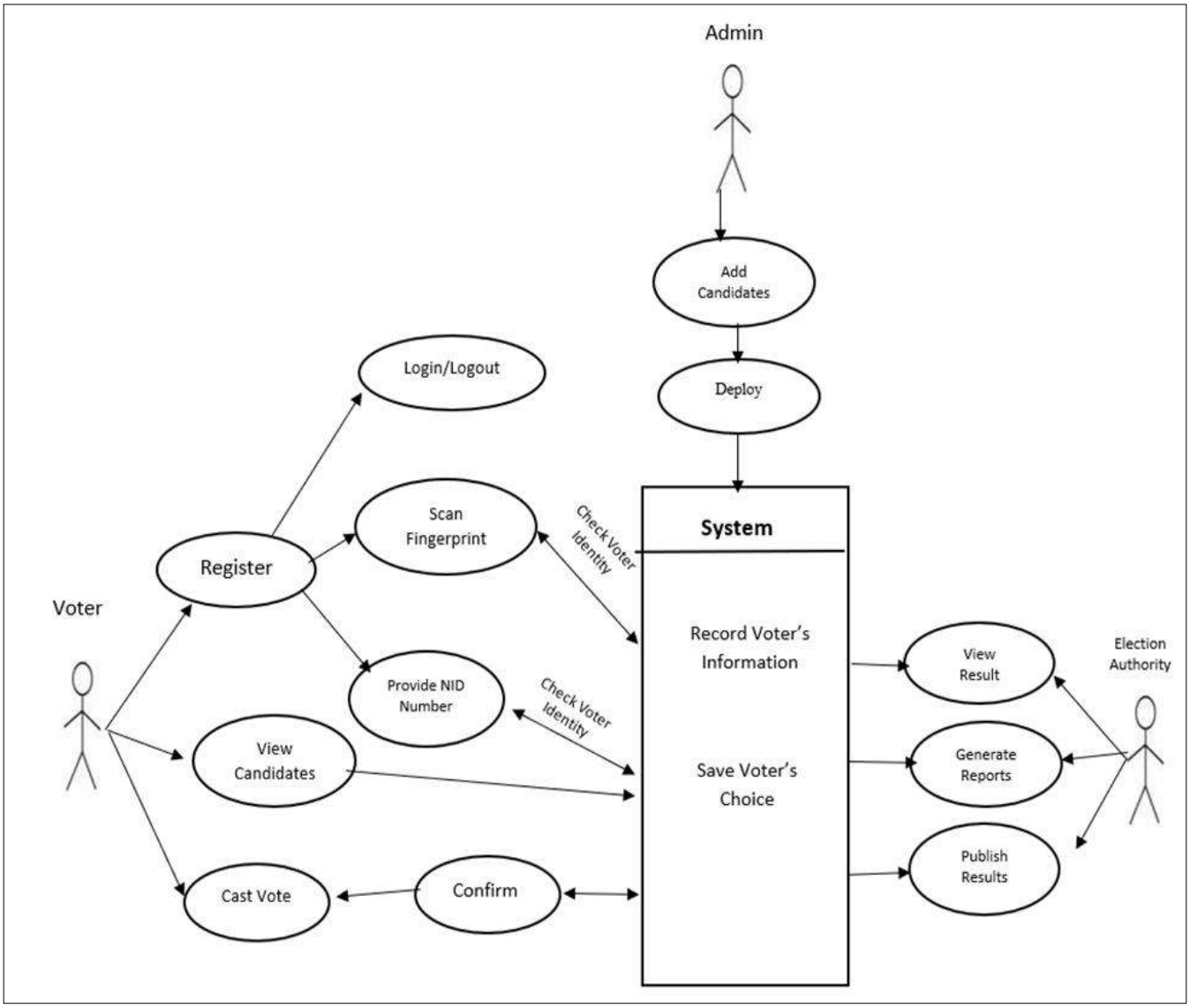

Fig 3: Use-Case-Diagram

Data Flow Diagram

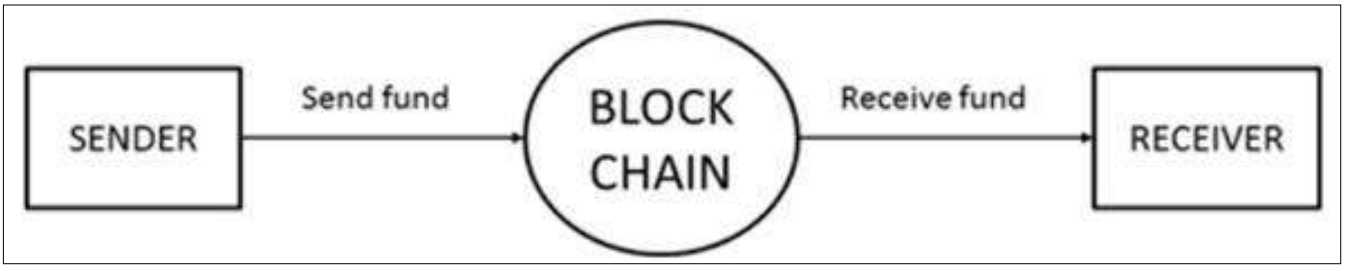

Fig 4: DFD Design

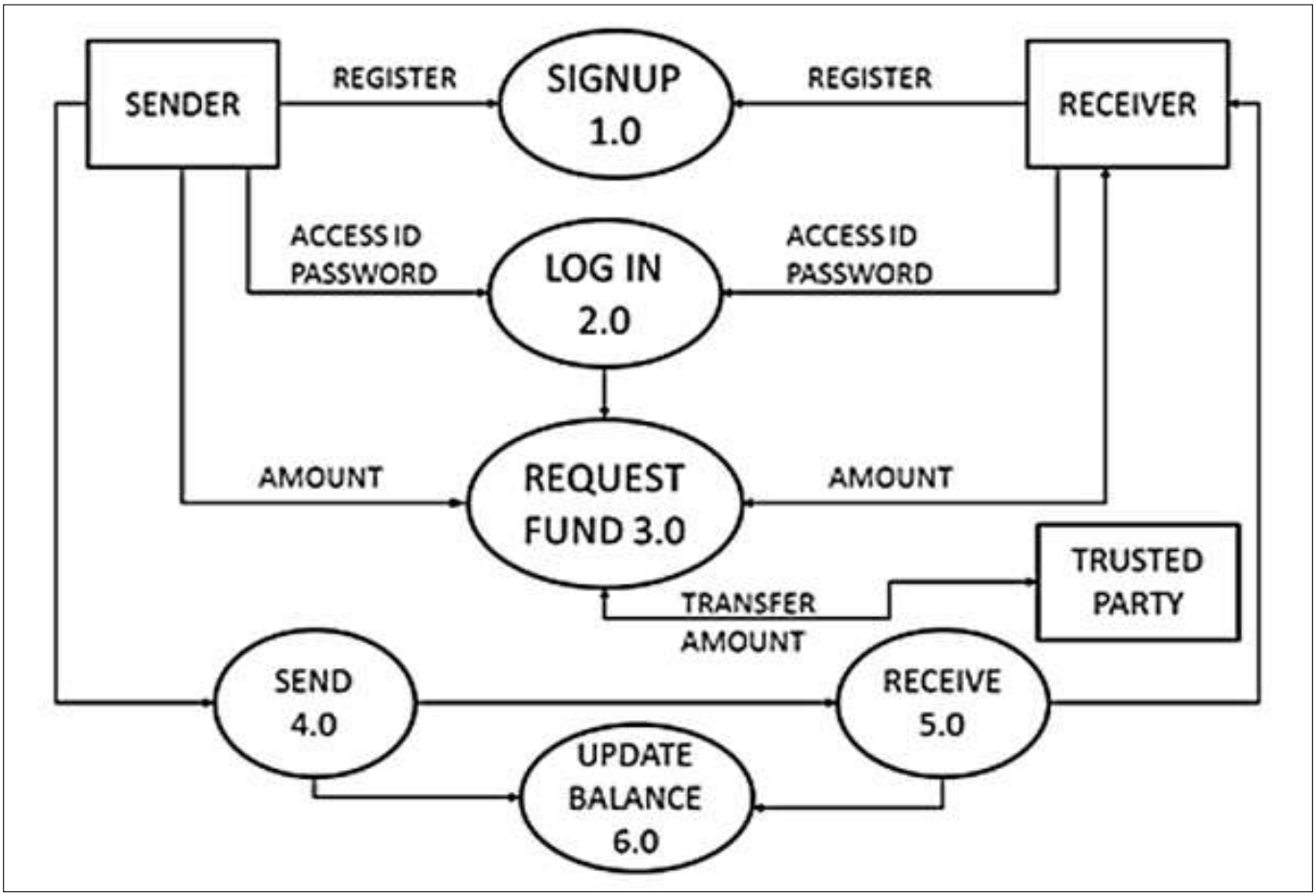

Fig 5: 1st Level DFD

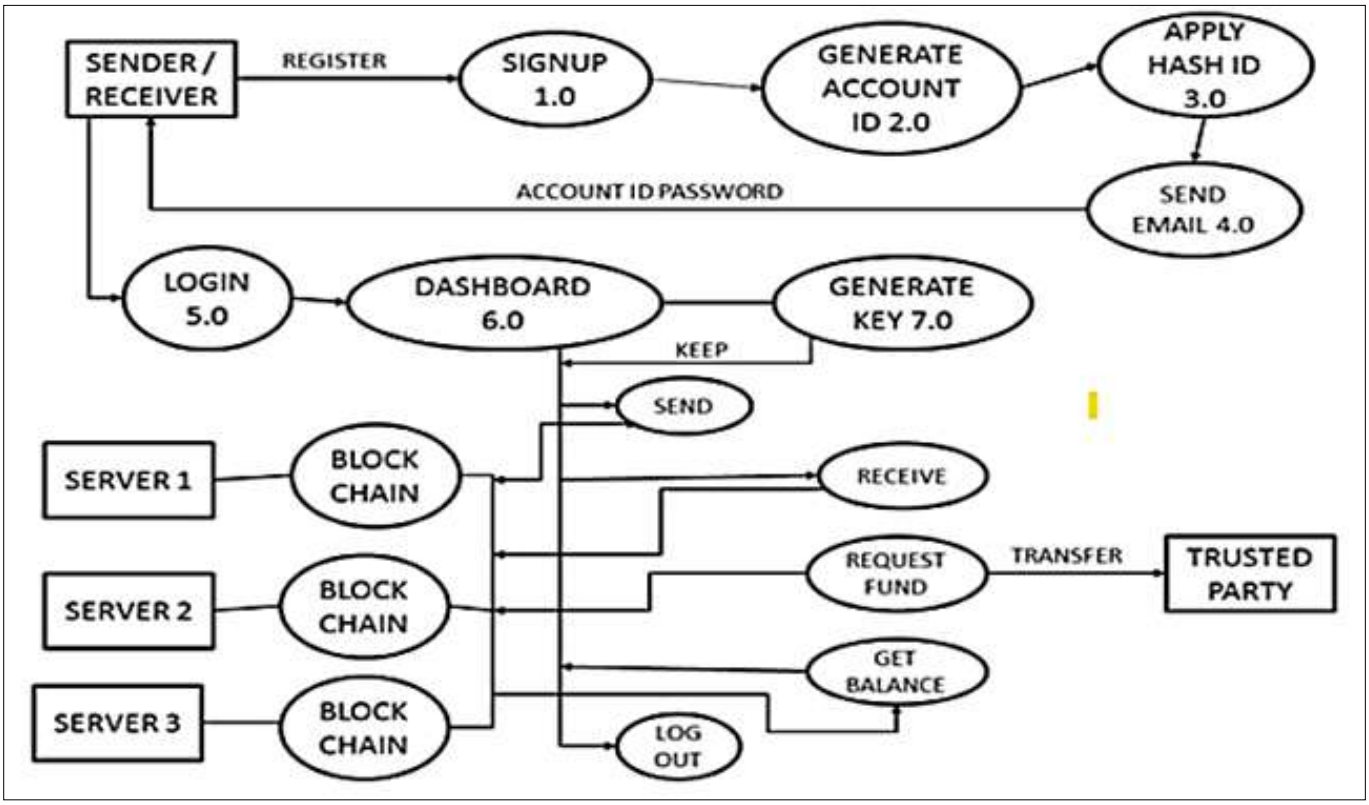

Fig 6: 2nd Level DFD

Project Description

Setting up: The first thing that we need to do is run local blockchain by starting up Ganache.

After setting up ganache there will be not any transaction as we have not done any transaction yet. As we can see from the snapshot below there is no transaction.

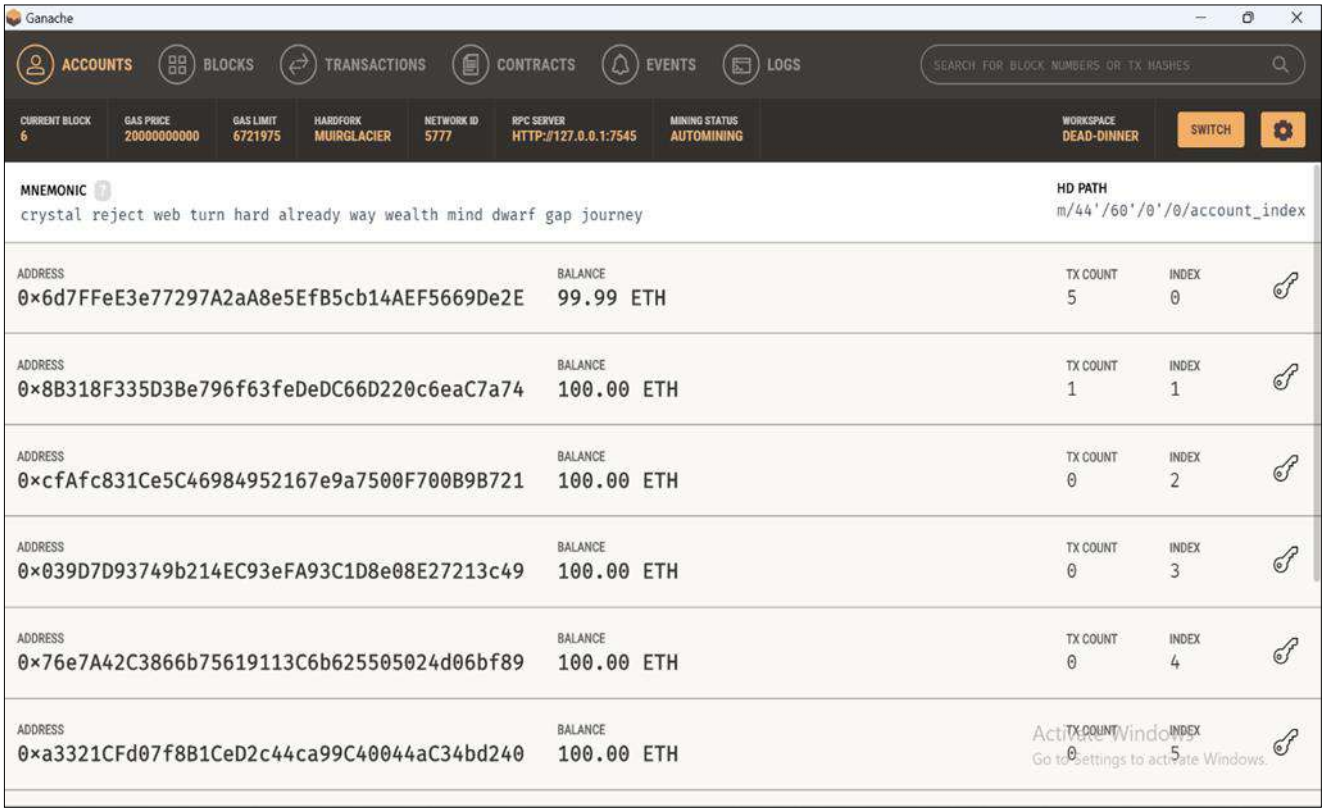

Fig 7: Screenshot of Setting up Ganache.

Now we use truffle framework to transfer the smart contract to the blockchain by giving command on the command line.

We have also used NPM directory by cmd. Following commands are being used for this purpose:

```
2_deploy_contracts.js
=====================

   Replacing 'Election'
   --------------------
   > transaction hash:    0x583300701f9979436ce409c7b2af99d87bb947dd5fa99cea4207fd329033bb1f
   > Blocks: 0            Seconds: 0
   > contract address:    0xb69ea8B4386fb2416A28b620d7747855E312C9DD
   > block number:        9
   > block timestamp:     1652538586
   > account:             0x6d7FFeE3e77297A2aA8e5EfB5cb14AEF5669De2E
   > balance:             99.9733302
   > gas used:            385673 (0x5e289)
   > gas price:           20 gwei
   > value sent:          0 ETH
   > total cost:          0.00771346 ETH


   > Saving migration to chain.
   > Saving artifacts
   -------------------------------------
   > Total cost:          0.00771346 ETH


Summary
=======
> Total deployments:   2
> Final cost:          0.01155232 ETH
```

Fig 8: Snapshot of Command Line for Truffle Framework.

After migrating the smart contract, we start the project using NPM directory by cmd.

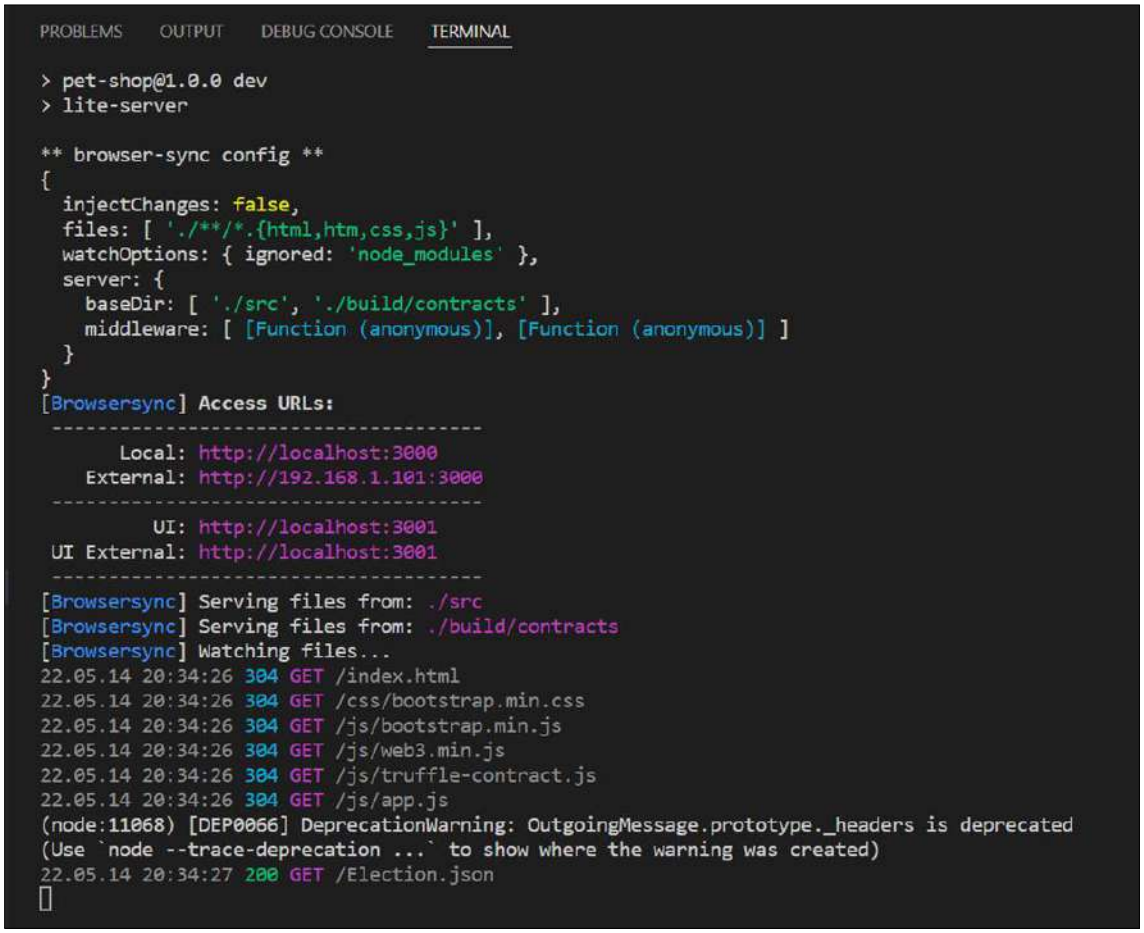

Fig 9: Command Line of NPM Directory

5.2 User Interface

User interface is through which users can interact with the e-voting system. The picture bellow is how user will see the interface. The loading screen will continue to display loading until the electorate login through metamask.

Below screen will be displayed when the electorate is logging in through metamask.

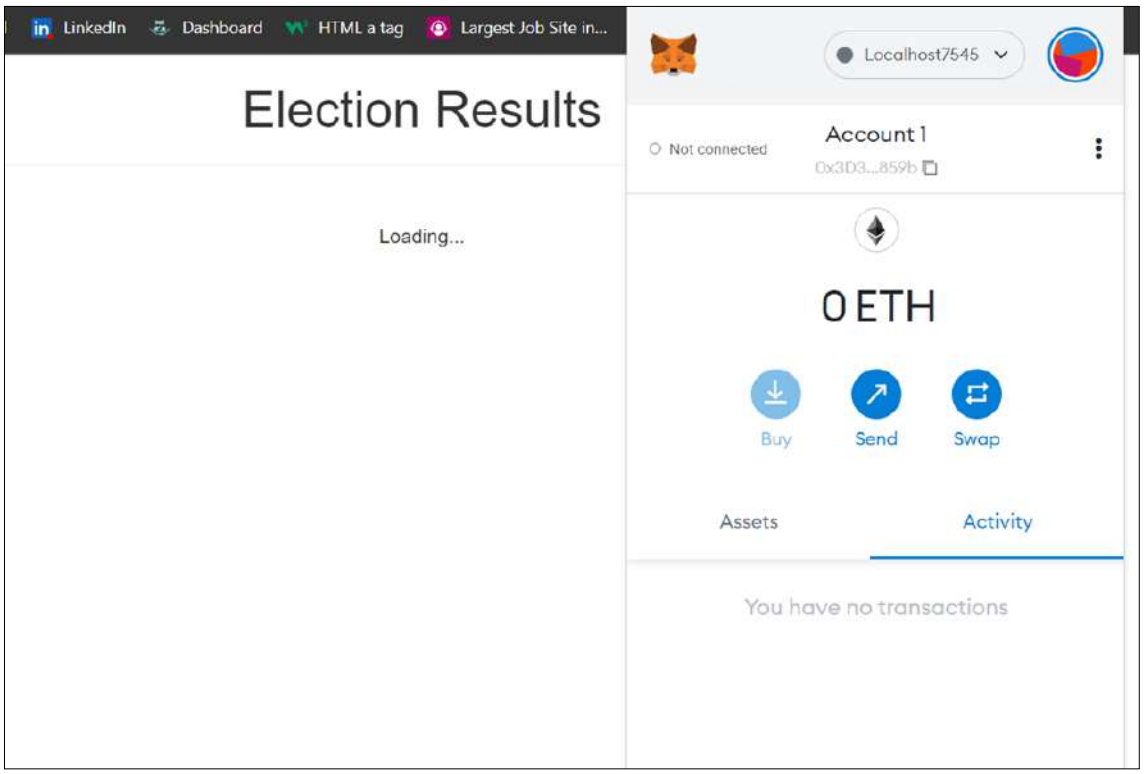

Fig 10: Constituency Logging in Via Metamask.

Fig 11: Main Screen

After the user has logged in, main screen comes up with zero vote, the user cannot vote until they import their account by entering private key.

Fig 12: Private Key.

Voter imports their account by entering the private key above.

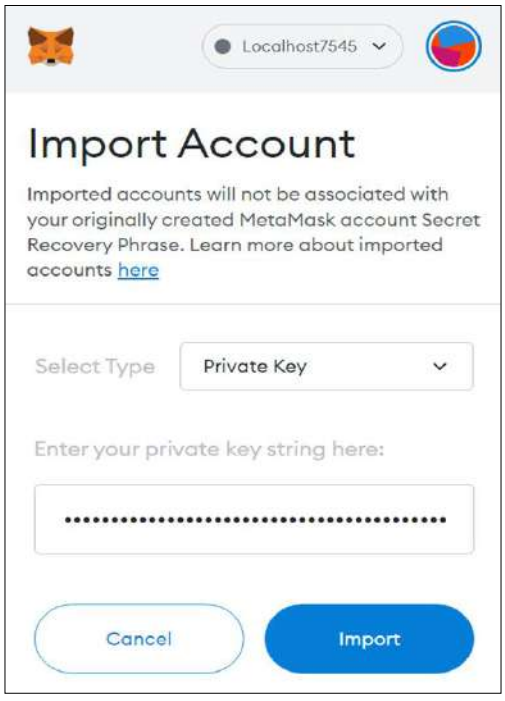

Fig 13: Snapshot of Importing Account

The electorates choose candidate of their choice, the metamask pop-up gets open when clicked on vote to confirm the transaction.

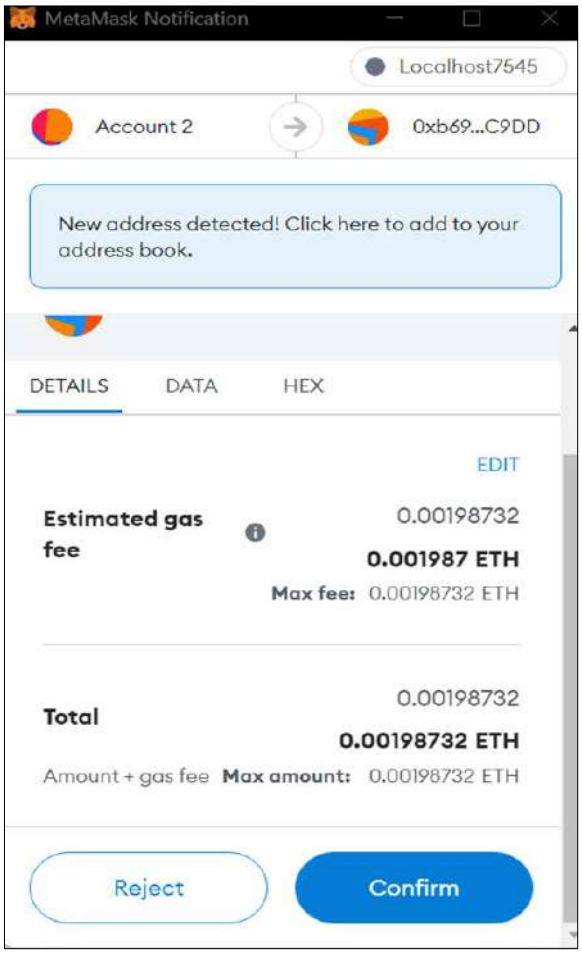

Fig 14: Snapshot of Confirming Transaction.

Once confirmation is done, the voter gets redirected to the main page where only results are visible but now you can't vote. In the similar manner, others can also vote by importing their account.

Fig 15: Snapshot of Result on Main Page.

Conclusion

The recent development in the area of the voting system includes Blockchain technology, which not only proved to be time and cost-efficient but is also safe and secure, hence is more reliable and precise than the earlier approaches. In this project, we will use blockchain-based e-voting using a smart contract that includes rules governing the communication and decision on the contract between parties. Various tools like Ganache, Truffle framework, NPM and Metamask were used for implementation as blockchain technology is decentralized. Tempering and alteration in such a system are quite attainable. Our proposed method provides convenience to the voters by allowing them to connect to an easy-to-use user interface system. They can cast their vote by importing their account and easily reviewing their vote. It creates a sense of trust among voters that their vote is being computed and kept in safe custody.

Limitation

Though this project has many advantages, it has some limitation as well, and this is listed Below-

- NID validator not used in the decentralize system.

Future Works

Anyone can develop the below mentioned features in future with their project-

- Validate the certificate-chain present in the NID and verify the eligibility of the voting citizen using a decentralized network of oracles.
- Extend support in the Dapp to Metamask alternatives such as Fortmatic.
- Store voting keys in the internal memory of the NID, in furtherance of convenience to the voting citizen.

References

1. Aranha DF, Ribeiro H, Paraense ALO. 'Crowdsourced integrity verification of election results', Annals of Telecommunications. 2016;71(7-8):1-11.
2. Ayed AB. 'A Conceptual Secure Blockchain Based Electronic Voting System', International Journal of Network Security & Its Applications (IJNSA). 2017;9(3):1-9.
3. Budurushi J, Renaud K, Volkamer M, Woide M. 'An investigation into the usability of electronic voting systems for complex elections', Annals of Telecommunications. 2016;71(7-8):309-322.
4. Elewa E, AlSammak A, Abd ElRahman A, ElShishtawy T. 'Challenges of Electronic Voting-A Survey', Advances in Computer Science: An International Journal. 2015;4(6):98-108.
5. Gjøsteen K, Lund AS. 'An experiment on the security of the Norwegian electronic voting protocol', Annals of Telecommunications. 2016;71(7-8):299-307.
6. Hsiao JH, Tso R, Chen CM, Wu ME. 'Decentralized E-Voting Systems Based on the Blockchain Technology', Advances in Computer Science and Ubiquitous Computing. CUTE 2017, CSA 2017. Lecture Notes in Electrical Engineering, vol 474. Springer, Singapore; c2018.
7. Jonath Alexander, Steven Lander, Ben Howerton. Netvote. A Decentralized Voting Network; c2018. Available at: https://netvote.io/wp content/uploads/2018/02/NetvoteWhite-Paper-v7.pdf
8. Neumann S, Volkamer M, Jurlind B, Prandrini M. 'Secivo: a quantitative security assessment model for internet voting schemes', Annals Telecommunication. 2016;71(7-8):337-352.